Intel® E7520 Memory Controller Hub (MCH)

Specification Update

June 2005

Notice: The Intel® E7520 MCH may contain design defects or errors known as errata that may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 303041-003

Contents

Revision History 4
Preface 5
Summary Table of Changes 6
Identification Information 9
Errata 10
Specification Changes 21
Specification Clarifications 22
Documentation Changes 23

Revision History

Version	Description	Date
-001	• Initial publication.	June 2004
-002	• Added C4 Stepping information. • Added errata 30-32.	November 2004
-003	• Added Specification Clarification 1.	June 2005

Preface

This document is an update to the memory interface specifications contained in the Affected Documents/Related Documents table below. This document is a compilation of device and document errata and specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the this update document and are no longer published in other documents. This document may also contain information that was not previously published.

Affected Documents/Related Documents

Document Title	Reference Number
Intel® E7520 Memory Controller Hub (MCH) Datasheet	303006

Nomenclature

Errata are design defects or errors. These may cause the Intel® E7520 MCH to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

Note: Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

Summary Table of Changes

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the E7520 MCH. Intel may fix some errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

X:	Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.
Doc:	Document change or update will be implemented.
Plan Fix:	This erratum may be fixed in a future stepping of the component.
Fixed:	This erratum has been previously fixed.
No Fix:	There are no plans to fix this erratum.
	Change bar to left of table row indicates this item is either new or modified from the previous version of this document.

Errata

No.	Stepping			Status	ERRATA
	C1	C2	C4		
1	X	X	X	No Fix	DMA channel source address checking error
2	X	X	X	No Fix	Data corruption after an illegal front side bus configuration Write
3	X	X	X	No Fix	Improper ECC and Memory Initialization while in Symmetric mode
4	X	X	X	No Fix	Single Channel ECC Error Injection issue
5	X	X	X	No Fix	PCI Express* add-in card presence detect state misreported
6	X	X	X	No Fix	Incorrect PCI Express Link/Lane numbers driven in degraded link
7	X	X	X	No Fix	PCI Express Compliance Mode issue
8	X	X	X	No fix	PCI Express Hot-Plug MSI interrupt issue
9	X	X	X	No Fix	PCI Express link training failures on hot reset
10	X	X	X	No Fix	Subsystem Identification and Subsystem Vendor Identification register issue
11	X	X	X	No Fix	MCH responds with illegal access on the Hub Interface for 32 GB configurations
12	X	X	X	No Fix	MCH hang on PCI Express enhanced configurations to non-existent devices causes hang
13	X	X	X	No Fix	Spurious errors logged during link training events
14	X	X	X	No Fix	DDR2 write offset issue
15	X	X	X	No Fix	DMA MSI interrupt issue
16	X	X	X	No Fix	SEC and DED error counters aliased in mirror mode
17	X	X	X	No Fix	HiLoCS bit not readable in memory error address registers
18	X	X	X	No Fix	MCH transitions from Polling.Active prematurely
19	X	X	X	No Fix	Non-fatal completion timeout errors observed on PCI Express devices
20	X	X	X	No Fix	SEC errors may be reported on opposite channel's error registers in memory mirroring mode
21	X	X	X	No Fix	MCH fails to train when non-TS1/TS2 training sequences are received
22	X	X	X	No Fix	DIMM sparing issue with demand scrub enabled
23	X	X	X	No Fix	Configuration transaction may be ignored in MCH when Configuration Request Retry Status is enabled in PCI Express to PCI/PCI-X bridges
24	X	X	X	No Fix	PCI Express Hot-Plug indicator blink causes extra SMBus write
25	X	X	X	No Fix	PCI Express x4, x8 links may train down to lower width
26	X	X	X	No Fix	SKP ordered set may not be sent within required interval
27	X	X	X	No Fix	END symbol omitted from the last PM_Request_Ack DLLP while entering L2 state on x1 PCI Express link
28	X	X		Plan Fix	System hang may occur when entering S4 and S5 power states
29	X	X		Plan Fix	Transposed interrupt messages across Hub Interface
30	X	X	X	No Fix	Completion timeout errors in the presence of heavy PCI Express peer-to-peer traffic
31	X	X	X	No Fix	SMBDAT and SMBCLK signals pulled down in S5
32	X	X	X	No Fix	Multiple PCI Express protocol errors may result in fatal receiver overflow

Specification Changes

Number	SPECIFICATION CHANGES
	None for this revision of the Specification Update

Specification Clarifications

Number	SPECIFICATION CLARIFICATIONS
1	Clarification to Section 4.4.1, “Memory Remapping”, in the EDS

Documentation Changes

Number	DOCUMENTATION CHANGES
1	Interupt Redirection

Identification Information

Component Identification via Programming Interface

The Intel® E7520 MCH can be identified by the following register contents:

MCH Version	Stepping	Vendor ID[1]	Device ID[2]	Revision Number[3]
E7520	C-1	8086h	3590h	09h
E7520	C-2	8086h	3590h	0Ah
E7520	C-4	8086h	3590h	0Ch

NOTES:
1. The Vendor ID corresponds to bits 15:0 of the Vendor ID Register located at offset 00 - 01h in the PCI bus 0, device 0, function 0 configuration space.
2. The Device ID corresponds to bits 15:0 of the Device ID Register located at offset 02 - 03h in the PCI bus 0, device 0, function 0 configuration space.
3. The Revision Number corresponds to bits 7:0 of the Revision ID Register located at offset 08h in the PCI bus 0, device 0, function 0 configuration space.

Component Marking Information

The Intel® E7520 MCH stepping can be identified by the following component markings:

MCH	Stepping	Spec
E7520	C-1	SL7P4
E7520	C-2	SL7RD
E7520	C-4	SL7XT

Errata

1. DMA channel source address checking error

Problem: In the DMA controller memory mapped registers, bit 6 of the DCRs (Descriptor Control Registers Memory Mapped I/O Address Offset 2Ch-2Fh, 6Ch-6Fh, 0ACh-A7h, 0ECh-EFh) for channels 0-3 should be RO, but is implemented as RW.

Implication: The DMA controller does not implement error checking for this case if this bit is set to “1”.

Workaround: Do not write a ‘1’ to bit 6 of the DCRx for channels 0-3.

Status: For the steppings effected, see the *Summary Table of Changes*.

2. Data corruption after an illegal front side bus configuration Write

Problem: When an illegal FSB configuration write occurs (bits [30:24] of the Configuration Address Register (CONFIG_ADDRESS, I/O address 0CF8h) are non-zero) PCI configuration accesses following this write may be corrupted.

Implication: This is a mishandled error case and causes corruption of transactions after this transaction. This is an illegal case.

Workaround: Do not write non-zero values to the PCI configuration address register reserved fields.

Status: For the steppings effected, see the *Summary Table of Changes*.

3. Improper ECC and Memory Initialization while in Symmetric mode

Problem: ECC and memory initialization is not properly executed when the MCH is in Symmetric Addressing mode. The MCH automatically enters symmetric address bit permuting when precisely four identical ranks of memory are available.

Implication: Correctable and uncorrectable memory errors may be detected since ECC is not properly initialized. The entire memory array is not initialized with zeros.

Workaround: Refer to your Intel representative for details

Status: For the steppings effected, see the *Summary Table of Changes*.

4. Single Channel ECC Error Injection issue

Problem: In single channel mode, single ECC error injection to Quad-word 4/5 or Quad-word 6/7 is not functional. The “Inject all” function works for all Quad-words as expected, as do all injection cases in dual channel mode.

Implication: Injected errors will not propagate to the memory array. As a result, when the memory location is read, the Correctable Read Memory Error Channel B and Correctable Read Memory Error Channel A of the DRAM_FERR Register (Device 0, Function 1, Offset 80h bit 0 and 8 Respectively) report no errors.

Workaround: Use “inject always” or limit error injection via the ECCDIAG register to the first half of the cache line when in single channel mode.

Status: For the steppings effected, see the *Summary Table of Changes*.

5. PCI Express* add-in card presence detect state misreported

Problem: PCI Express ports that are configured as non-hot plug capable incorrectly assert the add-in card Presence Detect State in the PCI Express Slot Status Register (EXP_SLTSTS Device 2-7, Function 0, Offset 7E-7Fh bit 6) regardless of the presence of an add-in card.

Implication: Software may interpret the presence of an add-in card when none exists.

Workaround: Utilize the Link Active bit in the Vendor Specific Status Register 1(VS_STS1 Device 2-7, Function 0, Offset 47h bit 1) as an alternative to the Presence Detect State bit.

Status: For the steppings effected, see the *Summary Table of Changes*.

6. Incorrect PCI Express Link/Lane numbers driven in degraded link

Problem: If a failure of receiver detect or bit/symbol lock occurs on lane 0 (lane 7 in the case of physical lane reversal) while other lanes successfully achieve bit/symbol lock in the early stages of Polling.Active, the MCH will exhibit anomalous lane numbering during the ensuing failed training sequence. Note that this anomalous behavior only occurs in situations where the combination of successful and failing lanes will result in a training failure, and a return to the Polling state.

Implication: When such a failed training is in progress, non-compliant non-PAD lane numbers may be observed on the MCH downstream lanes. The observed behavior may be seen as the MCH attempting a link split.

Workaround: None

Status: For the steppings effected, see the *Summary Table of Changes*.

7. PCI Express Compliance Mode issue

Problem: When a x8 link exits PCI Express Compliance Mode, the MCH will attempt to retrain as two x4 links. This issue manifests itself when the MCH inadvertently enters Compliance Mode.

Implication: Upon exiting Compliance Mode, the MCH link will attempt to train a downstream x8 device as two separate x4 links. Depending on the capabilities of the downstream device, the link width will be configured as either x4 or x1.

Workaround: Set bit 0 to 1b in Bus 0, Device 0, Function 0, Offset F5h. This will force the MCH to not enter compliance mode. Note that the MCH defaults to Compliance Mode disabled.

Status: For the steppings effected, see the *Summary Table of Changes*.

8. PCI Express Hot-Plug MSI interrupt issue

Problem: During a link down state, the MCH will not send MSI interrupts to the front side bus. In general MSI messages need not be delivered when the link is down, but in the event that MSI interrupt routing is used on Hot-Plug events, the processor will wait indefinitely for this interrupt. Waiting for command complete interrupts is a normal part of the steps in the orderly removal process, and link down will occur at the point that power is removed from the slot. Subsequent accesses to the slot control register to update indicators and power control will not generate the expected MSI interrupts from the MCH until slot power is restored, and the link is back up.

Implication: Hot-Plug software written to wait for command complete interrupts will hang in MSI interrupt mode.

Workaround: Run in either of the other two interrupt modes (the “legacy” method using the MCHGPE# to signal hot-plug interrupts to the ICH or “native” interrupt mode using PCI interrupts (INTA#)). Alternatively in MSI mode, software may poll for command complete rather than wait for MSI, or implement the command complete timeout to continue to the next slot control update rather than repeat the current slot control update.

Status: For the steppings effected, see the *Summary Table of Changes*.

9. PCI Express link training failures on hot reset

Problem: When issuing a hot reset via the bridge control register (BCTRL, Bus 0, Device 2-7, Function 0, Offset 3Eh bit 6, 1b) secondary bus reset bit to a PCI Express slot, the link may fall back degraded to a lower link width.

Implication: The link may degrade in width or fail to train all together after a hot reset.

Workaround: Implement a software algorithm that issues a Secondary Bus Reset upon a link training failure for 2 ms. The algorithm should support at least three iterations of Secondary Bus Resets.

Status: For the steppings effected, see the *Summary Table of Changes*.

10. Subsystem Identification and Subsystem Vendor Identification register issue

Problem: The Subsystem Vendor Identification register (SVID, Bus 0, D0:F0/F1, D1:F0, D2:F0 & D8:F0, Offset 2C-2Dh) and the Subsystem Identification register (SID, Bus 0, D0:F0/F1, D1:F0, D2:F0 & D8:F0, Offset 2E-2Fh) are not able to be written to independently. Writing to one register causes both to become Read Only.

Implication: If the values written to these two registers are not written via the Dword address, then the second value written will not be set.

Workaround: Write to both registers at the same time using PCI configuration Dword writes.

Status: For the steppings effected, see the *Summary Table of Changes*.

11. MCH responds with illegal access on the Hub Interface for 32 GB configurations

Problem: When devices behind the ICH try to access a memory address above 4 GB in systems with 32 GB of physical memory, an illegal access error is incorrectly flagged by the MCH.

Implication: A spurious error is flagged, and accesses between 4 GB and 32 GB will not succeed in the 32 GB (maximum) memory configuration, which can result in a system hang.

Workaround: Refer to your Intel representative for details the *Intel® E7520, E7320, and E7525 Memory Controller Hub (MCH) Components BIOS Specification* for details.

Status: For the steppings effected, see the *Summary Table of Changes*.

12. MCH hang on PCI Express enhanced configurations to non-existent devices causes hang

Problem: A system hang may occur when writing or reading to offsets above 0x0FF using the PCI Express enhanced configuration space of a non-existent device.

Implication: An invalid access error will be flagged, and a system hang may result.

Workaround: Polling or testing for devices must be done using offsets below 0x0FF. Access must not be issued to offsets above 0x0FF unless the targeted device is confirmed present.

Status: For the steppings effected, see the *Summary Table of Changes*.

13. Spurious errors logged during link training events

Problem: The MCH reports spurious receiver errors during initial link training, after a retrain, or after a secondary bus reset has occurred.

Implication: Spurious receiver errors will be logged in the associated port. There are no negative side effects besides the misreported error.

Workaround: Upon initial training and after each retrain or secondary bus reset, clear the correctable error detected bit of the PCI Express Device Status register (EXP_DEVSTS, Device 2-7, Function 0, Offset 6E-6Fh bit 0, 1b) and the receiver error status bit of the PCI Express Correctable Error Status register (EXP_CORERRSTS, Device 2-7, Function 0, Offset 110-113h bit 0, 1b). Also clear the FERR/NERR bits that flag correctable errors (EXP_FERR/EXP_NERR, Device 2-7, Function 0, Offset 160-163h / 164-167h bit 6, 1b).

Status: For the steppings effected, see the *Summary Table of Changes*.

14. DDR2 write offset issue

Problem: DQ/DQS signals terminate to a level about 300mv below VDDQ/2 between write bursts. No functional failures have been observed as a function of this issue.

Implication: Signal integrity issues may be observed.

Workaround: None

Status: For the steppings effected, see the *Summary Table of Changes*.

15. DMA MSI interrupt issue

Problem: If the MSI enable bit is cleared (disabled) in the MSI Control Register (MSICR - Device 1, Function 0, Offset B0-B3h bit 16, 0b) while DMA channels are active and generating interrupts, then corrupted MSI interrupts may be generated.

Implication: The MSI interrupt handler should not clear the MSI enable bit when DMA channels are in an active state or the system may hang due to the corrupted MSI.

Workaround: The following algorithm can be used to service MSI without having to disable them:

1. Check for MSI interrupt status. Check the DMA Controller Global Status Register (Device 1, Function 0, BAR 10h, Offset 104-107h bits 0-1, 8-9, 16-17, 24-25) to isolate the DMA channel generating the interrupt and then check the appropriate Channel Status Register for additional information about the interrupt (Device 1, Function 0, BAR 10h, Offsets 1C-1Fh, 44-47h, 80-83h & C4-C7).
2. Clear interrupt status in the Channel Status Registers for the respective channels.
3. Handle pending interrupts.
4. Exit if no interrupt status is set, else loop back to step 2 and repeat

   Once the handler has verified a read with no status set, it is safe to return, because any subsequent interrupt event will generate a new MSI, so the routine will need to be called again after it exits.

Status: For the steppings effected, see the *Summary Table of Changes*.

16. SEC and DED error counters aliased in mirror mode

Problem: During a read from DRAM in memory mirroring mode, the destination of the read (primary or mirrored DIMM) is dependent upon SA15, the state of the DDRCSR FSM Mirror State Transition Qualifier (Device 0, Function 0, Offset 9A-9Bh bits 11:10) and whether the access is a first or second attempt due to DED retry. If a correctable or uncorrectable error is encountered during a read, the primary DIMM's SEC or DED counter increments regardless of whether the primary or mirror DIMM was accessed. The mirror DIMM's counters do not increment in mirror mode.

Implication: In mirroring mode, primary DIMM SEC and DED error counters reflect the total number of errors across both the primary and mirror DIMMs.

Workaround: Refer to the *E7520, E7320 and E7525 BIOS Specification Update* for workaround details.

Status: For the steppings effected, see the *Summary Table of Changes*.

17. HiLoCS bit not readable in memory error address registers

In memory mirror mode, the Error Address registers utilize bit 0 to signal if the error occurred on the primary or mirror copy. In the MCH, these bits are not accessible via software and will always return 0b if read.The affected registers are:

Register	Device:Function:Offset
DRAM_SEC1_ADD	D0:F1:A0-A3h
DRAM_DED_ADD	D0:F1:A4-A7h
DRAM_SCRB_ADD	D0:F1:A8-ABh
DRAM_RETR_ADD	D0:F1:AC-AFh
DRAM_SEC2_ADD	D0:F1:C8-CBh

Implication: It is not possible to determine which of two DIMMs incurred an SEC or DED error with memory mirroring enabled.

Workaround: Refer to your Intel Representative for workaround details.

Status: For the steppings effected, see the *Summary Table of Changes*.

18. MCH transitions from Polling.Active prematurely

Problem: During a standard link training sequence, the MCH should remain in Polling.Active until TS1 ordered sets with link and lane set to PAD are received on all lanes that passed Receiver Detect. Because the MCH does not explicitly check for PAD on the link and lane numbers, it is possible for the MCH to transition from Polling.Active to Polling.Config when a downstream device is not executing a standard link training sequence (i.e. when the downstream device is actually in recovery or reset).

Implication: This early transition to Polling.Config may result in a degraded link width (e.g. a x4 port may train as x1), but the link will train.

Workaround: None required.

Status: For the steppings effected, see the *Summary Table of Changes*.

19. Non-fatal completion timeout errors observed on PCI Express devices

Problem: When PCI configuration accesses are made on secondary buses to MCH PCI Express bridges (Device 2-7, Function 0), non-fatal completion timeout errors (EXP_UNCERRSTS, Device 2-7, Function 0, Offset 104h bit 14) may be observed in the MCH. This condition also applies to PCI configuration accesses on any downstream device that is in the hot reset state or is disabled.

Implication: The system may escalate non-fatal PCI Express completion timeout errors inadvertently.

Workaround: There are two viable workarounds:

1. Mask the completion timeout errors on MCH PCI Express bridge devices with unpopulated slots as identified by the Present Detect State bit (EXP_SLTSTS, Device 2-7, Function 0, Offset 7Eh bit 6) in the PCI Express Slot Status register. If a device is present but disabled or in the hot reset state then the L ink Active bit (VS_STS1, Device 2-7, Function 0, Offset 47h bit 1) should be verified for link status.
2. Construct a completion timeout handler to clear the error and return if the Present Detect State bit and the Link Active bit are clear.

Status: For the steppings effected, see the *Summary Table of Changes*.

20. SEC errors may be reported on opposite channel's error registers in memory mirroring mode

Problem: In memory mirroring mode the MCH may report SEC errors on opposite channels depending on the state of SA15 and the DDRCSR FSM Mirror State Transition Qualifier (Device 0, Function 0, Offset 9A-9Bh bits 11:10). Channel A SEC errors may be reported in Channel B error registers and vice versa. Single bit errors occurring on Channel A would set bit 8 instead of bit 0 of both the DRAM First Error (DRAM_FERR, Device 0, Function 1, Offset 80-81h) and DRAM Next Error (DRAM_NERR, Device 0, Function 1, Offset 82-83h) registers. This also applies to respective SEC counter for the DIMMS in each channel. These registers are the DRAM_SEC_xx counters in Device 0, Function 1. The errors are reported correctly when the primary copy is read.

Implication: Errors may be misreported in opposite channel's error registers.

Workaround: Refer to the *E7520, E7320 and E7525 BIOS Specification Update* for workaround details.

Status: For the steppings effected, see the *Summary Table of Changes*.

21. MCH fails to train when non-TS1/TS2 training sequences are received

Problem: During the PCI Express training sequence, if a broken endpoint or a good endpoint on a broken board has correct receiver termination on any lane and transmits signals on that lane that can be seen at the MCH and are not valid TS1/TS2 training sequences, the MCH will fail to train that link at all.

Implication: The PCI Express specification intends that, if some lanes are transmitting bogus data instead of valid training sequences, those lanes should be treated as broken, and the link should fail down to an acceptable width (such as x1). If lane 0 were failing in this manner, the link would fail to train per the PCI Express specification. If a higher-numbered lane were failing in this manner, the PCI Express specification requires that the link attempt to train as a x1 on lane 0 - the MCH will not train in this scenario.

Failures are anticipated to occur because of a broken transmitter/receiver path, or a silent transmitter. None of those failure modes will cause the MCH to fail to train, since either the receiver termination will be missing, or the transmitted signals will not be seen at the MCH. In order to see invalid transmitted signals at the MCH, either a logic bug in the other PCI Express endpoint would be required, or a signal integrity issue so severe as to make operation impossible.

Workaround: None

Status: For the steppings effected, see the *Summary Table of Changes*.

22. DIMM sparing issue with demand scrub enabled

Problem: When spare copy is in progress and a demand scrub (as a result of a demand fetch with a correctable error) to an address resolving to the SCRUBLIM is performed, the process of spare copy from the failing DIMM to spare DIMM may terminate prematurely.

Implication: A system hang may occur when the spare DIMM is brought "on-line" prematurely and bad data is read from this DIMM. This condition is a result of the premature exit of the spare copy process.

Workaround: BIOS should disable demand scrub prior to initiating spare copy and re-enable it after the data migration is complete. Demand scrubbing can be enabled and disabled by updating the Scrub Limit and Control Register (SCRUBLIM Device 8, Function 0, Offset C8-CBh bit 27).

Status: For the steppings effected, see the *Summary Table of Changes*.

23. Configuration transaction may be ignored in MCH when Configuration Request Retry Status is enabled in PCI Express to PCI/PCI-X bridges

Problem: Under certain circumstances that include a mix of PCI Express traffic in the presence of completions with Configuration Retry Status (configuration space traffic receiving CRS, and other traffic that is posted / governed by Posted Flow Control credits) on a given PCI Express port, the MCH may ignore and fail to issue an outbound configuration space access indefinitely. This behavior has been observed in configurations with PCI Express to PCI/PCI-X bridge devices under circumstances where at least one device “behind” the bridge is active and operational, while at least one other device “behind” the bridge remains unresponsive to configuration requests for an extended period of time. Such failures ultimately manifest themselves as CPU IERR# assertions, which commonly precipitates a platform reboot. Completions with Configuration Request Retry Status are generally sent by a PCI Express to PCI/PCI-X bridge when it relays configuration space traffic to a PCI/PCI-X device which exhibits a long latency in responding to configuration space traffic. The CRS completion status mechanism is intended to prevent a PCI Express completion timeout from occurring in cases where historical PCI/PCI-X implementations would experience an extended latency without response, but would not generate any timeout or associated error.

Implication: A system hang may occur.

Workaround: To avoid configuration transactions from being ignored, Intel strongly recommends that BIOS should disable Configuration Request Retries in all PCI Express bridge devices. For Intel® 6700PXH 64-bit PCI Hub this is accomplished by clearing the Bridge Configuration Retry Enable bit in the Device Control register (D0:F0,2:R04Ch bit 15). This bit is cleared by default. Some PCI or PCI-X devices may require lengthy self-initialization sequence (up to 1.5 sec as defined by PCI Express Base Specification 1.0a) to complete before they are able to service Configuration Requests after reset. In order to ensure the ability of the system to successfully enumerate PCI devices, BIOS should disable PCI Express Completion Timeout in the root port configuration of MCH links connected to Intel® 6700PXH 64-bit PCI Hub, Intel® IOP332, and Intel® 41210 devices (including add-in cards) by setting the Completion Timeout Timer Disable bit in the Vendor Specific command register (D2-7:F0:R045h bit 3). BIOS should ensure that the Completion Timeout Timer remains enabled (default) for other active PCI Express links. BIOS should also ensure that the Completion Timeout Error Mask is set in MCH root ports associated with inactive PCI Express links (unpopulated slots or disabled devices) -- refer to erratum 19 for detail.

Status: For the steppings effected, see the *Summary Table of Changes*.

24. PCI Express Hot-Plug indicator blink causes extra SMBus write

Problem: When both PCI Express device 4 and 6 are configured for Hot-Plug functionality, an attention or power indicator blink command sent to the I/O Expander on device 4 will cause an extra SMBus write to the external I/O expander connected to device 4 and 6. An attention or power indicator blink command on device 6 will not generate extra SMbus write command for device 4. No failures have been observed from this erratum.

Implication: Extra writes may be observed on the SMbus that have no side effects.

Workaround: None

Status: For the steppings effected, see the *Summary Table of Changes*.

25. PCI Express x4, x8 links may train down to lower width

Problem: It has been observed that x4, x8 links may fail to train to their full link widths. This behavior occurs infrequently. The issue is caused by the MCH exiting the Polling.Active state and entering the Polling.Config state prior to the downstream device entering the Polling.Active state.

Implication: PCI Express ports may fail train to at full width.

Workaround: Intel recommends an algorithm that will issue an Secondary Bus Reset upon a link training failure for 2ms. The algorithm should support at least three iterations of Secondary Bus Resets.

Status: For the steppings effected, see the *Summary Table of Changes*.

26. SKP ordered set may not be sent within required interval

Problem: During Link Recovery on a PCI Express port, the MCH may fail to transmit a SKP ordered set within the required time interval as defined in the PCI Express 1.0a Specification if a TLP or DLLP was pending when the link entered Recovery.Idle state.

Implication: If the receiving device depends upon receipt of a SKP ordered set to progress through Link Recovery, a timeout will occur resulting in Link Down and automatic reinitialization of the PCI Express link. A link transitions through Recovery only under exceptional operational conditions. Following the Link Recovery timeout and reinitialization, the link should resume normal operation unless the original Link Recovery condition was entered as a result of a hard failure mechanism.

Workaround: None

Status: For the steppings effected, see the *Summary Table of Changes*.

27. END symbol omitted from the last PM_Request_Ack DLLP while entering L2 state on x1 PCI Express link

Problem: When a x1 link transitions into the L2 state, the MCH may fail to transmit the END symbol of the last PM_Request_Ack DLLP.

Implication: If a downstream device expects an END symbol in the last PM_Request_Ack DLLP from the MCH, it may incorrectly decode the electrical ordered set that follows. Endpoints should expect the COM symbol in the electrical ordered set to indicate a final confirmation to transition the link to the L2 state.

Workaround: None

Status: For the steppings effected, see the *Summary Table of Changes*.

28. System hang may occur when entering S4 and S5 power states

Problem: When the system is transitioning into the S4 or S5 state, the MCH may fail to respond to an ICH power management handshake event resulting in a system lock. Specifically, when the duration between the rising edge of HICLK and the preceding rising edge of HCLKIN is between 1.6ns - 2.7ns when measured at the MCH pins, it is possible to encounter this erratum. This also implies that if a platform is outside this range, this erratum will not be encountered.

When this failure occurs the system will maintain power and remain unresponsive indefinitely. Once the system becomes unresponsive after encountering this erratum, it will only resume operation after an AC power cycle or an unconditional powerdown.

Under normal operation, a transition into S3-S5 will have the following processor bus signature:

1. ICH asserts STPCLK# to the processor.
2. Processor issues a Stop Grant Acknowledge transaction on the processor bus.
3. ICH asserts SLP# to the processor.

In the failing case steps 1 and 2 are observed, but step 3 is not.

Implication: System may hang during a power management transition.

Workaround: Refer to your Intel Representative for workaround details.

Status: For the steppings effected, see the *Summary Table of Changes*.

29. Transposed interrupt messages across Hub Interface

Problem: In cases where virtual wire interrupt messages (Assert/Deassert-INT[A, B, C, D]) received on PCI Express are spuriously short (the deassert message is received before the assert message can be forwarded by the MCH to the ICH), the MCH may infrequently transpose the interrupt assert and deassert messages across the Hub Interface. Under normal conditions the MCH will forward an interrupt assert message originating from a PCI Express port over the Hub Interface prior to receiving or forwarding the corresponding deassert message. In the event that the transposition occurs, the virtual wire is left asserted at the ICH when it is in fact de-asserted at the source. The virtual wire will remain asserted until a subsequent interrupt on that same virtual wire arrives to clear the condition. During the period where the virtual wire remains "stuck" asserted, spurious interrupts will be forwarded to the processor(s).

Implication: Systems running with a single logical processor (most commonly in uni-processor configurations when Hyper-Threading Technology is disabled) and operating in legacy PIC mode or virtual wire mode A may hang under high I/O-driven interrupt stress. For systems operating in full APIC mode where the number of virtual interrupt lines (intA, intB, etc.) used by all PCI Express adapters in a system exceeds the number of logical processors (threads), the system may hang.

Workaround: BIOS updates are required to support PIC mode. Use of PCI Express adapters is not recommended. PCI Express devices down on the motherboard are supported if they are single function devices or have their IOAPIC enabled. Refer to your Intel representative for details.

Status: For the steppings effected, see the *Summary Table of Changes*.

30. Completion timeout errors in the presence of heavy PCI Express peer-to-peer traffic

Problem: When a single PCI Express port receives a continuous stream of posted transactions targeting a peer PCI Express port (as opposed to targeting memory), and the throughput into the sending port is equal or higher than that of the destination port, the MCH will continuously grant the sending port access to the target port until a break in the posted traffic occurs. Under these conditions, a third PCI Express port attempting to send one or more posted transactions to the same target port will be held off for an unbounded period of time (until a break occurs in the transmit stream from the port currently granted access). Given the right mix of traffic to the port that is thus blocked, and sufficient duration on the "continuous stream" of posted transactions at the target port, a completion timeout error may occur on the port that is blocked. Note that outbound CPU traffic to the target port and completions for inbound reads from the target port are not impacted by the blocking mechanism; only competing peer transfers to the target will be stalled.

Implication: When PCI Express peer-to-peer transfers are sufficiently large and uninterrupted, and transfers are initiated on multiple source ports targeting the same destination port, completion timeout errors may occur. In order to trigger such a timeout, one of the peer source ports must be blocked for at least 16.7 ms.

Workaround: Limit the uninterrupted duration (total data payload size) for transfers between peer PCI Express ports, such that no one continuous transfer will exceed a duration of 16.7 ms. For reference, each x4 PCI Express port is capable of transferring well over 12 MB of data in 16.7 ms, thus an uninterrupted blockage of such duration is not expected to occur unless extreme circumstances are contrived.

Status: For the steppings effected, see the *Summary Table of Changes*.

31. SMBDAT and SMBCLK signals pulled down in S5

Problem: According to *SMBus Specification 2.0* the SMBDAT and SMBCLK signals are to float while in the S5 state. Due to device protection circuitry these signals are pulled down while in the S5 state.

Implication: Devices on auxiliary power such as a BMC that share an SMBus connection with the MCH will not be able to signal on the SMBus in the S5 state due to the signals being pulled down.

Workaround: A mux can be incorporated into the SMBus to disconnect the MCH when the platform goes into the S5 state.

Status: For the steppings effected, see the *Summary Table of Changes*.

32. Multiple PCI Express protocol errors may result in fatal receiver overflow

Problem: If a PCI Express device connected to the MCH generates multiple transaction layer protocol errors, including, unexpected completion packets or malformed transaction layer packets (TLPs) that otherwise pass all link-layer error checking, and have the correct alignment on the interface, the MCH may experience a fatal receiver overflow.

Implication: If the above conditions are met, The MCH may detect and log a “fatal” receiver overflow error. MCH behavior in the presence of this error is consistent with the specification, in that continued operation on the port after such an error may be unreliable.

Workaround: Intel recommends avoiding use of PCI Express devices that generate unexpected completion or malformed TLP protocol violations. If this is unavoidable, the receiver overflow error detected by the MCH may be escalated to a system event (e.g.: SERR#) that prevents continued operation on the compromised link.

Status: For the steppings effected, see the *Summary Table of Changes*.

33. System marginalities may result in spurious link-down error events on power state changes

Problem: On system power state changes (S3, S4, and S5) PCI Express devices are placed in the D3 device power state by the operating system, which results in automatic negotiation with the MCH to enter the L1 link state. In systems where the cumulative noise present at the MCH receiver pins exceeds the MCH receiver threshold for detecting Electrical Idle, the transition into L1 may fail to complete normally, ultimately resulting in a spurious link-down error from the MCH. If link down error (D2-7:F0:R140h, bit 11) is escalated using a fatal system error (SERR#) mechanism, a blue-screen may result on exposed systems.

The PCI Express specification for Electrical Idle at the receiver is 65 mV peak-peak differential, and characterization of the MCH indicates that some lanes on some devices are marginal with respect to this specification. While L1 failures should be exceedingly rare, Intel recognize that this specification is difficult to meet, and acknowledge the exposure

Implication: Systems with sufficient noise at the MCH receivers and a BIOS profile that escalates the “link down error” as a fatal system event may be exposed to blue-screen occurrence on system power state transitions. Exposure to the error increases with the cumulative noise (platform noise + silicon noise) present at the MCH receivers when the link is in Electrical Idle. Systems utilizing a BIOS configuration that does not escalate the “link down error” as a fatal error are not exposed.

Custom operating systems or future operating systems that independently manage the power state of PCI Express devices outside the scope of system power state transitions would be similarly exposed to link-down errors via the same mechanism. In cases where the destination power state on the attached device is between D0 and D3, any such link-down event constitutes a real error from which software may only recover by fully reconfiguring the devices below the affected link.

Workaround: None

Status: For the steppings affected, see the *Summary Table of Changes*.

34. Possible loss of Hot-swap Power Fault Event in dual PCI Express Hot-swap port configurations

Problem: During boot, as part of normal PCI enumeration, the external hot-swap expander device on the PCI Express hot-swap ports must be configured. This PCI enumeration proceeds on a per device basis,

during which an expander input change on the second port might get lost. There is sufficient time between configuration of the first PCI Express hot-swap port and the second port for this erratum to happen, due to interaction between the internal hot-swap controller and the external hot-swap expander.

Implication: A power fault event, as an example, could occur on the second port before it was configured. The power fault event is thus not reported.

Workaround: Force the configuration of both hot-swap controllers to occur back-to-back in time. This prevents any controller/expander traffic other than the configuration until both expanders have been configured, and ensures that the controller and the external expander are in agreement. In BIOS, make sure that setting the hot-swap capable bit for one of the hot-swap ports is followed immediately by setting the same bit for the other hot-swap port.

Status: For the steppings affected, see the *Summary Table of Changes*.

Specification Changes

There are no Specification Changes in this revision of the Specification Update.

Specification Clarifications

1. Clarification to Section 4.4.1, “Memory Remapping”, in the EDS

Section 4.4.1 currently reads as follows:

4.4.1 Memory Remapping

An incoming address (referred to as a logical address) is checked to see if it falls in the memory remap window. The bottom of the remap window is defined by the value in the REMAPBASE register. The top of the remap window is defined by the value in the REMAPLIMIT register. An address that falls within this window is remapped to the physical memory starting at the address defined by the TOLM register.

A clarification will be made to Section 4.4.1 by adding a second paragraph which will read as follows:

4.4.1 Memory Remapping

An incoming address (referred to as a logical address) is checked to see if it falls in the memory remap window. The bottom of the remap window is defined by the value in the REMAPBASE register. The top of the remap window is defined by the value in the REMAPLIMIT register. An address that falls within this window is remapped to the physical memory starting at the address defined by the TOLM register.

The remap operation increases the latency of CPU to memory accesses (within the remap area) by three clocks when the pipeline between the CPU interface and memory is empty (the “idle latency” case). This may result in a measurable performance degradation within the remap range for latency-sensitive benchmarks that are run on lightly loaded systems. This latency difference disappears when latency-sensitive benchmarks are run on moderately to heavily loaded systems.

Documentation Changes

There are no Documentation Changes in this revision of the Specification Update.

1. Interupt Redirection

The bit definition for the hardware interrupt redirection has been added. The following changes will be reflected in the next release of the Datasheet.

REDIRCTL - Redirection Control - (D8:F0)
Address Offset: 4C - 4Fh
Access: R/W, RO
Size: 32 bits
Default Value: 0000_648Ch

Bit Field	Default & Access	Description
31:14	00001h	Reserved
13	1b R/W	**Interrupt Redirection Algorithm (XTPR).** 0 = LRU (least recently used within the lowest priority pool) 1 = highest number in lowest priority pool, default
12:0	048Ch	Reserved